

## Game AI Competitions: An Open Platform for Computational Intelligence Education

### Introduction

Teaching computational intelligence (CI) to undergraduate and/or graduate students is challenging because the theories are difficult and they feel that the topics are not closely related to their lives. It is desirable to use interesting projects to attract students' attention into the computational intelligence. CI educators have been used different types of tools to define course projects for students. There are no standard course materials on the projects, hence each instructor develops his/her own projects by oneself, making it difficult to share them. Furthermore, if the projects are dependent on special hardware or software, it often becomes a bottleneck on the dissemination of the materials.

When a CI educator plans to use games for education, there are lots of difficult decisions and it requires much effort to realize them. For example, they need to find games suitable for CI techniques and then they need to confirm that the AI can be programmed into them. The next step is to define goals of the projects and objective measure to evaluate the goodness of the students' works. In this step, it is necessary to adjust the difficulty of the problems not to lose their interest in the projects. If you have more time, you need to design your own solution checking whether the problem is feasible or not. After then, you finally upload a sample solution that can be a starting point for students.

The game AI competitions have been a part of several international conferences related to AI and CI techniques. For example, IEEE Conference on Computational Intelligence in Games (IEEE CIG) has hosted a lot of interesting game AI competitions since 2005 (<http://www.ieee-cig.org/>). In 2005, the competitions covered IPD (Iterated Prisoners Dilemma), Go (7×7), and computer controlled car racing. For many years, the competitions have been evolved to include a lot

of different games with new challenges (For example, Turing test [1] and game contents generation). Table 1 summarizes the game AI competitions offered from several international conferences. Their advancement from many participants has been reported in the main competition website and the international conferences/journals [2]. The competitions have been a good benchmarking tool to evaluate the goodness of new approaches for different genres of games [3].

**TABLE 1** Summary of game AI competitions.

GAME	COMPETITIONS	MAIN WEBSITES
MS PAC-MAN	MS PAC-MAN VS GHOSTS COMPETITION	<a href="http://www.pacman-vs-ghosts.net/">HTTP://WWW.PACMAN-VS-GHOSTS.NET/</a>
	MS PAC-MAN COMPETITION (SCREEN-CAPTURE VERSION)	<a href="http://cswww.essex.ac.uk/staff/sml/pacman/pacmancontest.html">HTTP://CSWWW.ESSEX.AC.UK/STAFF/SML/PACMAN/PACMANCONTEST.HTML</a>
SUPER MARIO	GAME PLAY TRACK	<a href="http://www.marioai.org/">HTTP://WWW.MARIOAI.ORG/</a>
	LEARNING TRACK	
	LEVEL GENERATION TRACK	
	TURING TEST TRACK	
SUPER TUX	LEVEL GENERATION TRACK	<a href="http://platformersai.com/">HTTP://PLATFORMERSAI.COM/</a>
	TURING TEST TRACK	
SIMULATED CAR RACING	SIMULATED CAR RACING CHAMPIONSHIP	<a href="http://games.ws.dei.polimi.it/competitions/scr/">HTTP://GAMES.WS.DEI.POLIMI.IT/COMPETITIONS/SCR/</a>
UNREAL TOURNAMENT	2K BOTPRIZE	<a href="http://www.botprize.org/">HTTP://WWW.BOTPRIZE.ORG/</a>
STARCRAFT	IEEE CIG	<a href="http://ls11-www.cs.uni-dortmund.de/rtts-competition/">HTTP://LS11-WWW.CS.UNI-DORTMUND.DE/RTS-COMPETITION/</a>
	AIIDE	<a href="http://www.aiide.org/starcraft">HTTP://WWW.AIIDE.ORG/STARCRAFT</a>
	PHYSICAL TSP LEAGUE	<a href="http://www.ptsp-game.net/">HTTP://WWW.PTSP-GAME.NET/</a>
PHYSICAL TSP [4]	MULTI-OBJECTIVE PTSP	
GEOMETRY FRIENDS	THE AI COOPERATION GAME COMPETITION	<a href="http://gaips.inesc-id.pt/geometryfriends/">HTTP://GAIPS.INESC-ID.PT/GEOMETRYFRIENDS/</a>
GO	HUMAN VS. COMPUTER GO COMPETITION	<a href="http://oase.nutn.edu.tw/WCCI2012/">HTTP://OASE.NUTN.EDU.TW/WCCI2012/</a>

# 2014 IEEE Symposium Series on Computational Intelligence

## IEEE SSCI 2014



December 9 to 12, 2014, Orlando, Florida, U.S.A.



### Important dates:

**Paper submission:** June 15, 2014  
**Notification to authors:** September 5, 2014  
**Final submission:** October 5, 2014  
**Early registration:** October 5, 2014  
**Website:** [www.ieee-ssci.org](http://www.ieee-ssci.org)

### Organizing Committee

#### Honorary Chairs:

Marios Polycarpou, Cyprus  
David Fogel, USA

#### General Chair:

Haibo He, USA

#### General Program Chair:

Cesare Alippi, Italy

#### Finance Chair:

Gary Fogel, USA

#### Keynote and Tutorial Chair:

Derong Liu, China

#### Local Arrangement Chair:

Fred Ham, USA

#### Best Paper Award Chair:

Gary Yen, USA

#### Publication Chairs:

Robert Kozma, USA

Nian Zhang, USA

Zhigang Zeng, China

#### Registration Chairs:

Yaochu Jin, UK

Zeng-Guang Hou, China

#### Publicity Chairs:

Hisao Ishibuchi, Japan

Danil Prokhorov, USA

Plamen Angelov, UK

Kay Chen Tan, Singapore

Chin-Teng (CT) Lin, Taiwan

#### Special Sessions Chair:

Robi Polikar, USA

#### Poster Session Chair:

Dongbin Zhao, China

#### Doctoral Consortium Chairs:

Demetrios G. Eliades, Cyprus

Stephen G. Matthews, UK

#### Website Chair:

Li-Wei (Leo) Ko, Taiwan

Welcome to the sunshine Orlando, Florida for the IEEE SSCI 2014, a flagship international conference sponsored by the IEEE Computational Intelligence Society (CIS) promoting all aspects of Computational Intelligence (CI). The IEEE SSCI 2014 co-locates multiple exciting symposiums at one single location, providing a unique opportunity to encourage cross-fertilization and collaborations in all areas of CI. The IEEE SSCI 2014 features a large number of keynotes, tutorials, and special sessions. The IEEE SSCI 2014 will also offer a number of travel grants as well as an exciting Doctoral Consortium. We hope you could participate this exciting event, and look forward to seeing you in Orlando in December 2014!

### IEEE SSCI 2014 Symposia

**ADPRL'14**, IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, Huaguang Zhang, China, Jagannathan Sarangapani, USA, Lucian Busoniu, France.

**CCMB'14**, IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain, Robert Kozma, USA, Leonid Pervlovsky, USA, Damien Coyle, UK.

**CIASG'14**, IEEE Symposium on Computational Intelligence Applications in Smart Grid, G. Kumar Venayagamoorthy, USA, Jung-Wook Park, Korea.

**CIBIM'14**, IEEE Symposium on Computational Intelligence in Biometrics and Identity Management, Qinghan Xiao, Canada, David Zhang, China, Fabio Scotti, Italy

**CICA'14**, IEEE Symposium on Computational Intelligence in Control and Automation, Xiao-Jun Zeng, UK.

**CICARE'14**, IEEE Symposium on Computational Intelligence in healthcare and e-health, Amir Hussain, UK.

**CIComms'14**, IEEE Symposium on Computational Intelligence for Communication Systems and Networks, Sergey Andreev, Finland, Raymond Carroll, Ireland, Maode Ma, Singapore, Paolo Rocca, Italy.

**CICS'14**, IEEE Symposium on Computational Intelligence in Cyber Security, Dipankar Dasgupta, USA

**CIDM'14**, IEEE Symposium on Computational Intelligence and Data Mining, Zhi-Hua Zhou, China, Barbara Hammer, Germany, Carlotta Domeniconi, USA, Lipo Wang, Singapore.

**CIDUE'14**, IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments, Yaochu Jin, UK, Robi Polikar, USA, Shengxiang Yang, UK

**CIES'14**, IEEE Symposium on Computational Intelligence for Engineering Solutions, Michael Beer, UK, Rudolf Kruse, Germany, Vladik Kreinovich, USA.

**CIHLI'14**, IEEE Symposium on Computational Intelligence for Human-like Intelligence, Jacek Mandziuk, Poland, Wlodzislaw Duch, Poland.

**CIMSIVP'14**, IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing, Khan M Iftikharuddin, USA.

**CISDA'14**, IEEE Symposium on Computational Intelligence for Security and Defense Applications, Nur Zincir-Heywood, Canada, Akira Namatame, Japan, Rafael Falcon, Canada.

**EALS'14**, IEEE Symposium on Evolving and Autonomous Learning Systems, Plamen Angelov, U.K., Dimitar Filev, USA, Nikola Kasabov, New Zealand.

**FOCI'14**, IEEE Symposium on Foundations of Computational Intelligence, Manuel Ojeda-Aciego, Spain.

**IA'14**, IEEE Symposium on Intelligent Agents, Hani Hagras, UK, Vincenzo Loia, Italy.

**ICES'14**, IEEE International Conference on Evolvable Systems, Andrew Tyrrell, UK, Pauline Haddow, Norway

**CIR2AT'14**, IEEE Symposium on Computational Intelligence in Rehabilitation and Robotic Assistive Technologies, Gui DeSouza, USA, James Patton, USA, Georgios Kouroupetroglou, Greece

**MC'14**, IEEE Workshop on Memetic Computing, Zexuan Zhu, China.

**MCDM'14**, IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making, Yaochu Jin, UK, Piero Bonissone, USA, Juergen Branke, UK

**RiSS'14**, IEEE Workshop on Robotic Intelligence in Informationally Structured Space, Janos Botzheim, Japan, Chu Kiong Loo, Malaysia.

**SDE'14**, IEEE Symposium on Differential Evolution, Janez Brest, Slovenia, Swagatam Das, India, Ferrante Neri, UK

**SIS'14**, IEEE Symposium on Swarm Intelligence, Yuhui Shi, China, P. N. Suganthan, Singapore.



**FIGURE 1** Game AI competition portal (<http://cilab.sejong.ac.kr/gc>).

Although the competitions have been designed to accelerate the research of the CI in Games community, they are also recognized as a good tool for education. Without significant changes, the competitions can be easily transformed into course projects. There are lots of benefits to use the competitions for the education. As the competitions are open to public, they provide all kinds of materials on the preparation of the competitions. Organizers carefully choose games for the CI-based competitions. Usually, they adopt open-source games and relatively inexpensive commercial games (US\$10~20). You need to buy game software for StarCraft and unreal tournament competitions.

Organizers have developed additional source codes (defined as API), sample programs and documents written to support participants making their own AI codes. Also, they define the problems to be solved and the measure to rank the entries. Finally, some of the competitions (usually continued for several years) provide source codes of participants including the winner of the year. This is a good starting point for the students. Also, the participants often publish papers on their approaches to international conferences and journals that are useful for in-depth study. In addition to the benefits, educators can encourage students to submit their own entry to the international competitions and if possible, technical papers (IEEE CIG has a special track dedicated to the competition papers).

## A Portal Site for Game AI Competitions

From 2011, game AI competition portal (Figure 1) compiled by authors and stu-

dents have been used to support the courses with the game AI competitions. The portal was designed to be the starting point of the game AI competitions for research and education. For each competition, the portal contains information on the rules, step-by-step instructions to start the first project, historical archives (video, results, and source codes of the entries), and educational resources (video tutorials, links and slides). Since 2012, the portal has been used as course supplementary materials for the students' course projects.

This section briefly introduces the preparation of the game AI competitions (for an instance, simulated car racing). We recommend that the readers visit the game AI competition portal and the simulated car racing section. In the section, you can access the summary of rules and the easy-to-follow, step-by-step instructions to run the first sample code from the organizers. From our experience, inexperienced students wasted much time to complete the first step of the "entry making." If we can help them to reduce the technical burden in the early-stage of development, they can concentrate on their AI module.

After you finish the task to run the first sample, you are ready to build your own entry. However, it is desirable to understand the fundamental ideas widely used in the competition before you start your own project. At first, you are recommended to follow the video tutorials offered at the portal which explains the important concepts in a step-by-step manner with videos. For the beginners, it is safe to keep the simulated car in the middle of the track and adjust car

speed based on the difference between target and current one. The tutorial provides an example of source code implementation and videos of results for the basic skills.

You can start your own controller by modifying the sample controller (SimpleDriver.cpp) provided by organizers (Figure 2). This class implements basic functions for a general car racing controller. You can easily modify each function with your own idea to enhance the performance of your controller. You can also download the source codes of participants from competition websites as it is better use them rather than starting from scratch.

In the historical archive section of the portal, you can review the results and winner of the previous year's competition. In the section, you can download the source codes (or executables) of the others' entries on the competition websites. Reviewing the source codes of the entries will give you great amount of insights on your controller design. For a better understanding of the codes, it is recommended to read technical papers and documents published by authors (especially, IEEE CIG competition track papers). You can feel the state-of-the art of the competition from the analysis of the entries (especially the winner of the competitions).

From the resource section of the portal, you can easily find some relevant papers on the competition, it is necessary to investigate the state-of-the-art used for the competition. For the simulated car racing, Fuzzy, NN (Neural Network), and rules have been widely used as the representation of knowledge. Several optimization techniques have been adopted to tune the parameters of the controllers (for example, evolutionary computation (EC) and swarm intelligence). Figure 3 shows the distribution of the techniques used in the car racing competitions (from the survey of 80 papers). It shows that EC, Fuzzy, NN and ENN (Evolutionary NN) are dominant techniques (76%) for the problem. With the data, you can determine your own methods or create novel approach to improve the performance.



<pre> // SimpleDriver.cpp int SimpleDriver::getGear(CarState &amp;cs){     // Your own codes on gear change     return gear; } float SimpleDriver::getSteer(CarState &amp;cs){     // Your own codes to decide steering     return Steer; } float SimpleDriver::getAccel(CarState &amp;cs){     // Your own codes to decide acceleration and brake     return accel_and_brake; } CarControl SimpleDriver::wDriver(CarState cs){     // Your own codes to detect stuck     if(Stuck){ // Your own codes for recovering policy}     else { float accel_and_brake = getAccel(cs);         int gear = getGear(cs);         float steer = getSteer(cs); }     clutching(cs,&amp;clutch);     CarControl cc(accel, breake, gear, steer, clutch);     return cc; } </pre>	<pre> // CarState.h class CarState{     float getAngle();     void setAngle(float angle);     float getCurLapTime();     void setCurLapTime(float curLap Time);     float getDamage();     void setDamage(float damage);     float getDistFromStart();     void setDistFromStart(float disFromStart);     float getDistRaced();     void setDistRaced(float distRaced);     ....     void setTrackPos(float trackPos);     float getWheelSpinVel(int i);     void setWheelSpinVel(int i, float value);     float getZ();     void setZ(float z); } </pre>
--	---

**FIGURE 2** Modifying client source codes to implement your idea (car racing).

## Game AI Competitions and Education

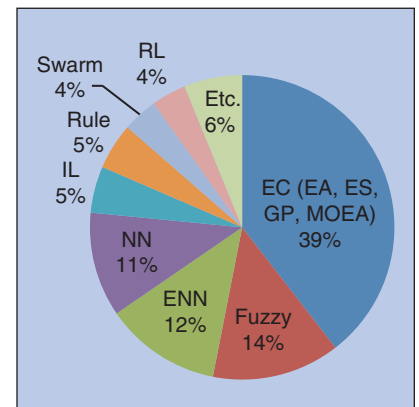
Recently, there have been some AI and CI courses offered with the game AI competitions<sup>1</sup>. The idea is straightforward and students can design and implement a computer program for the game AI competition as an assignment or project for the course. As an assignment, students are asked to program well-known (known to be successful for the game) algorithms into the games. From this experience, students can learn the state-of-the-art techniques for video and real-time strategy games. As a project, students are recommended to propose a novel method for the game AI competition and implement their ideas. Because the competitions are designed as benchmarking tools, it is easy to evaluate the goodness of their novel approaches.

□ **Game AI (2012, Mark Riedl, Georgia Tech):** In the course, the lecturer dealt with all the important

materials on the game AI design (Procedural Contents Generation, decision making, path planning, and so on). He adopted the Ms Pac-Man vs Ghosts Competition and Super Mario (Playing and Level Generation Track). He slightly changed the rules of the competition and hosted internal competitions to give evaluation scores.

□ **Modern AI for Games (2012, Julian Togelius, IT University of Copenhagen):** In the course, the lecturer adopted the game AI competition for individual assignments and group projects. In the lecture, he taught tree search (Monte-Carlo), evolutionary computation, neural networks and reinforcement learning. In the individual assignments, he asked students to implement one of the four major techniques from the classes into the Ms Pac-Man vs Ghosts competition. Local competitions were done during the course. The group projects were based on student's novel idea for Car Racing simulation and Super Mario.

□ **Agents AI & Games (2011, Saturnino Luz, Trinity College Dublin):** In this lecture, he taught several interesting topics on agents (architecture



**FIGURE 3** Percentage of techniques used in the car racing competition (RL = reinforcement learning, IL = imitation learning).

and types of agents) and machine learning (reinforcement learning and optimization algorithms). In the preparation of the course, he assigned a survey of each game AI competition to students. After their survey, the final competition project was determined as Ms Pac-Man vs. Ghost.

Authors have attempted to open the courses with the game AI competitions since 2009. S.-B Cho offered a course to graduate students with the game AI competitions. During the course, teams of students submitted their own controllers for three internal small-scale

<sup>1</sup>Game AI (Mark Riedl, Georgia Tech) <https://research.cc.gatech.edu/inc/cs-4731-spring-2012>. Game AI (Michael Mateas, UC Santa Cruz) <http://courses.soe.ucsc.edu/courses/cm146/Winter12/01/pages/syllabus>. Modern AI for Games (Julian Togelius, IT University of Copenhagen) <https://blog.itu.dk/MAIG-E2012/>. Agents, AI & Games (Saturnino Luz, Trinity College Dublin) <https://www.scss.tcd.ie/~luzs/t/cs7032/>.



**TABLE 2 Courses offered by authors with the game AI competitions.**

	YEAR	STUDENTS (# OF STUDENTS)	COMPETITIONS	NOTE
SOFTWARE AGENTS	2009	GRADUATE (28)	CAR RACING, MS PAC-MAN AND UNREAL TOURNAMENT	
C PROGRAMMING	2009	1ST YEAR UNDERGRADUATE (40)	CAR RACING	
WINDOWS PROGRAMMING	2010	3RD YEAR UNDERGRADUATE (10)	CAR RACING, MS PAC-MAN	
ADVANCED ARTIFICIAL INTELLIGENCE	2011	GRADUATE (10)	CAR RACING, STARCRAFT	GAME AI COMPETITION PORTAL WAS BUILT
ARTIFICIAL INTELLIGENCE	2012	GRADUATE (17)	CAR RACING, SUPER MARIO, UNREAL TOURNAMENT	GAME AI COMPETITION PORTAL WAS USED

competitions (car racing, Ms Pac-Man and unreal tournament games). He provided with materials of the basic introduction, techniques used, and how to start each competition. The final grade was completely dependant on the ranks of teams in the three competitions.

K.-J. Kim offered a “C Programming” course for freshmen in the spring semester of 2009. In the course, the freshman studied the grammar of the “C programming language.” Traditionally, the final project of the course had been a set of “well-defined text-based” programming problems. However, they are not sufficient to motivate young generation familiar with fantastic graphics. Car racing competition was offered as an alternative of the final course project. As the competition software was written mainly in C++, it is not suitable for students in the “C Programming” course. He wrote a sample controller that students could program without knowing C++ language. Although students designed their controllers only with rule-based approaches, the projects gave them ideas on game AI [5].

He also offered a “Windows Program Design” course for the junior students in the spring semester of 2010. In the course, there was minimum teaching and all the time was spent for course projects in computer labs. One team chose the car racing competition and applied imitation learning to generate car racing controllers. They generated training samples using the 2009 champions and trained decision trees to control several actuators of cars. The second team also used the car racing competition software

for their projects and applied several ideas to build their controllers. They tried to build controllers that drive as safe as possible and adopted several ideas: Remembering crash points and decreasing speed before corners. The last team used Ms Pac-Man and combined the ideas of the last year’s winner with the basic software platforms provided by organizers. The entries (car racing and Ms Pac-Man) in this class were submitted to IEEE CIG 2011.

Since 2011, we have attempted to construct the game AI competition portal for the purpose of education. It can save much time to introduce the main concept of the game AI competitions. After general overview in the class, students can start their own project from the instructions on the portal. With the help of the portal, our students can submit their own entries to the IEEE CIG 2011 (3rd rank in StarCraft competition) and GECCO 2012. Also, they published international journal papers based on their course reports [6][7]. At WCCI (World Congress on Computational Intelligence) 2012 and WASA (Workshop at SIGGRAPH ASIA) 2012, we gave tutorials to introduce the game AI competitions based on the portal. Recently, we got several requests to use the portal for graduate courses from researchers.

### Concluding Remarks

It is not easy to run competitions in courses because it needs customized software and well-defined rules. This is one of the reasons that teachers avoid the competitions as their educational tools. Fortunately, the game AI competitions are publicly opened and you can easily

modify the competition software for your favor. Our students are familiar with games and they are willing to build their own games. This is one of popular reasons that students choose computer engineering as their major. If you use games as materials of education, it is easy to open your students’ minds on your teaching subjects. Initially, students start to build their own controller using several rules with hand-tuned parameters. They suddenly realize the limitations of their approach and naturally think about computational intelligence. Now, they are transformed into highly motivated young researchers for computational intelligence.

### Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2013 R1A2A2A01016589, 2010-0018948).

### References

- [1] P. Hingston, “A turing test for computer games bots,” *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 3, pp. 169–186, 2009.
- [2] P. Rohlfshagen and S. M. Lucas, “Ms Pac-Man Versus Ghost Team CEC 2011 competition,” in *Proc. IEEE Congr. Evolutionary Computation*, 2011, pp. 70–77.
- [3] S. Karakovsky and J. Togelius, “The Mario AI benchmark and competitions,” *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 55–67, 2012.
- [4] D. Perez, P. Rohlfshagen, and S. M. Lucas, “The physical travelling salesman problem: WCCI 2012 competition,” in *Proc. World Congr. Computational Intelligence*, 2012, pp. 1–8.
- [5] K.-J. Kim and S.-B. Cho, “Experience on running a small-size simulated car racing tournament in an introductory programming course,” in *Proc. Int. Conf. Convergence Hybrid Information Technology*, 2010, pp. 200–206.
- [6] K.-J. Kim, J.-H. Seo, J.-G. Park, and J.-C. Na, “Generalization of TORCS car racing controllers with artificial neural networks and linear regression analysis,” *Neurocomputing*, vol. 88, pp. 87–99, July 2012.
- [7] T.-S. Kim, J.-C. Na, and K.-J. Kim, “Optimization of autonomous car controller using self-adaptive evolutionary strategy,” *Int. J. Adv. Robot. Syst.*, vol. 9, pp. 31–45, 2012.

